RU2148856

Publication Title:

INFORMATION EXCHANGE SYSTEM

Abstract:

Abstract of RU 2148856

(C1) Translate this text FIELD: computer engineering, in particular, portable data processing units. SUBSTANCE: device has at least one portable data processing unit, which has data transmission units, data processing unit and memory unit, which has first region with control program, and second region, which stores descriptions of various connection modes between data processing units represented as interaction contexts. EFFECT: optimal constraints imposed by memory unit size, protected loading of software codes. 17 cl, 5 dwg

---

(19) **RU** (11) **2 148 856** (13) **C1**

(51) МПК[7] **G 06 F 15/16, G 07 F 7/10, G 06 K 19/07**

РОССИЙСКОЕ АГЕНТСТВО
ПО ПАТЕНТАМ И ТОВАРНЫМ ЗНАКАМ

(12) ОПИСАНИЕ ИЗОБРЕТЕНИЯ К ПАТЕНТУ РОССИЙСКОЙ ФЕДЕРАЦИИ

(21), (22) Заявка: **96118111/09, 08.02.1995**

(24) Дата начала действия патента: **08.02.1995**

(30) Приоритет: **08.02.1994 EP 94200236.1**

(46) Дата публикации: **10.05.2000**

(56) Ссылки: **EP 0466969 A2, 22.01.92. DE 4126213 A1, 11.02.93. WO 87/07062 A1, 19.11.87. EP 0190733 A3, 13.08.86. SU 1615731 A2, 23.12.90. SU 1820392 A1, 07.06.93.**

(85) Дата перевода заявки PCT на национальную фазу: **08.09.1996**

(86) Заявка PCT:
NL 95/00055 (08.02.1995)

(87) Публикация PCT:
WO 95/22126 (17.08.1995)

(98) Адрес для переписки:
129010, Москва, ул. Б.Спасская 25, стр.3, "Городисский и Партнеры", Емельянову Е.И.

(71) Заявитель:
БЕЛЛЕ ГАТЕ ИНВЕСТМЕНТ Б.В. (NL)

(72) Изобретатель: **ДЕ ЙОНГ Эдуард Карел (NL)**

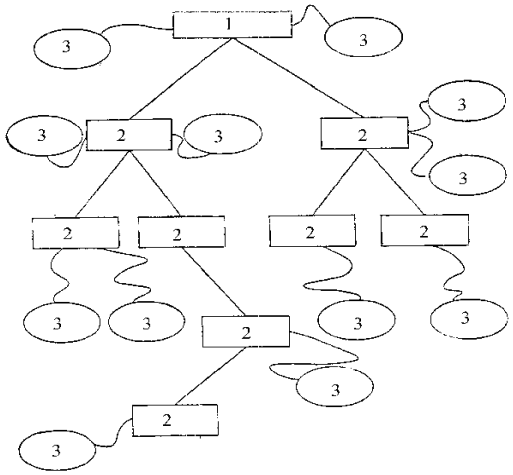
(73) Патентообладатель:
БЕЛЛЕ ГАТЕ ИНВЕСТМЕНТ Б.В. (NL)

(74) Патентный поверенный:
Емельянов Евгений Иванович

(54) СИСТЕМА ИНФОРМАЦИОННОГО ОБМЕНА

(57) Реферат:

Изобретение относится к системам информационного обмена с портативными блоками обработки данных. Техническим результатом является создание средства, предназначенного для оптимизации ограничений, налагаемых размерами области памяти, и создание механизма защищенной загрузки программных кодов. Система информационного обмена состоит по меньшей мере из одного портативного блока обработки данных, содержащего в себе средство передачи данных, средство обработки и запоминающее средство, причем последнее содержит первую область с управляющей программой и вторую область, содержащую описания возможных режимов связи между блоками обработки данных в виде контекстов взаимодействия. 16 з.п. ф-лы, 5 ил.



Фиг.1

RU 2148856 C1

RU 2148856 C1

(19) **RU** (11) **2 148 856** (13) **C1**

(51) Int. Cl.[7] **G 06 F 15/16, G 07 F 7/10, G 06 K 19/07**

RUSSIAN AGENCY
FOR PATENTS AND TRADEMARKS

(12) ABSTRACT OF INVENTION

(21), (22) Application: 96118111/09, 08.02.1995

(24) Effective date for property rights: 08.02.1995

(30) Priority: 08.02.1994 EP 94200236.1

(46) Date of publication: 10.05.2000

(85) Commencement of national phase: 08.09.1996

(86) PCT application:
NL 95/00055 (08.02.1995)

(87) PCT publication:
WO 95/22126 (17.08.1995)

(98) Mail address:
129010, Moskva, ul. B.Spasskaja 25, str.3,
"Gorodisskij i Partnery", Emel'janovu E.I.

(71) Applicant:
BELLE GATE INVESTMENT B.V. (NL)

(72) Inventor: DE JONG Ehduard Karel (NL)

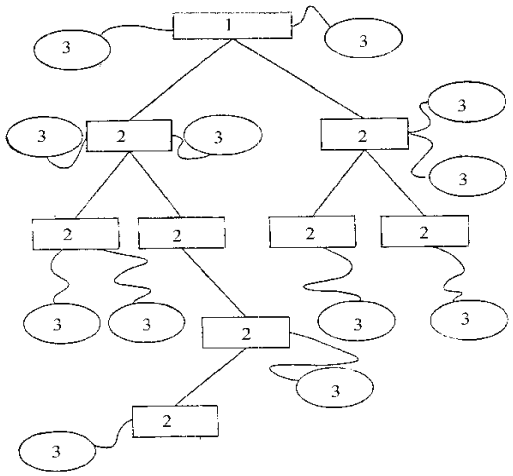
(73) Proprietor:
BELLE GATE INVESTMENT B.V. (NL)

(74) Representative:
Emel'janov Evgenij Ivanovich

(54) INFORMATION EXCHANGE SYSTEM

(57) Abstract:

FIELD: computer engineering, in particular, portable data processing units. SUBSTANCE: device has at least one portable data processing unit, which has data transmission units, data processing unit and memory unit, which has first region with control program, and second region, which stores descriptions of various connection modes between data processing units represented as interaction contexts. EFFECT: optimal constraints imposed by memory unit size, protected loading of software codes. 17 cl, 5 dwg



Фиг.1

RU 2148856 C1

RU 2148856 C1

Изобретение касается системы информационного обмена, включающей в себя по меньшей мере один портативный блок обработки данных, содержащий средство передачи данных, средство обработки данных и запоминающее устройство, причем последнее содержит в себе управляющую программу.

Такая система известна из международной заявки на патент WO-A-87/07063, в которой описана система, прданазначенная для портативного носителя информации, имеющего файлы многочисленных прикладных задач. Одним из наиболее важных применений такого портативного носителя информации является интеллектуальная карточка, пригодная для многочисленных применений. Известный носитель информации описан как носитель иерархических структурированных данных с характеристиками защиты для обеспечения решения многочисленных прикладных задач с одним и тем же носителем данных. Прикладные задачи представлены в виде наборов данных. В указанной заявке описано выполнение системы иерархических файлов на носителе информации для запоминания изменяемых данных в сочетании с иерархическим набором разрешений доступа. Носитель информации реагирует на набор общих команд. Разрешения доступа к файлу различаются для различных операций и допускаются в зависимости от подтверждения пароля. Вводится счетчик попыток верификаций пароля, а также предусмотрено уничтожение запомненных данных в качестве санкции против слишком большого количества попыток обращений.

Известный носитель информации представляется, в первую очередь, в виде запоминающего устройства, а не в виде процессора. С помощью управляющей программы типа двоичной логической операции можно выполнять только очень простые функции. Не допускается выполнение непредусмотренного набора операций по запросу терминала, устанавливающего связь с носителем информации. Единственную возможность защиты обеспечивает введение верификации пароля. В известной системе невозможны другие проверки условий доступа. Помимо этого, каждая прикладная задача для носителя информации имеет свой собственный файл в памяти носителя информации. Отсутствуют специальные меры повышения эффективности располагаемой области памяти, которая, особенно на интеллектуальных карточках, весьма ограничена и, следовательно, устанавливает ограничения на количество возможных применений.

Патент EP-A-0479655 касается осуществления проверок условий доступа в интеллектуальных карточках. Однако для раскрытой здесь технологии желательно обеспечить меры для введения возможности других проверок условий доступа.

Патент EP-A-0361491 касается системы программирования плат с микросхемами, обеспечивающей осуществление защитного перепрограммирования карточек. В нем описано использование режимов доступа с однократной записью для управления доступом к частям программируемой памяти. Таким образом, можно расширить количество применений одной карточки. Описана проверка условий доступа различными методами, включая криптографические протоколы.

Патент EP-A-0292248 касается загрузки прикладных программ в интеллектуальную карточку, используя неизменяемую программу операционной системы. Она включает в себя осуществление способа форсирования режима доступа к данным, используя зоны памяти с назначенными атрибутами доступа. Особые режимы доступа представляют собой "однократную запись" (которая описана только косвенно) и "только выполнение".

Патент США 4874935 касается программирования карточек с использованием словаря данных, описывающего формат элементов данных, запомненных в памяти карточки. Словари данных обычно отличаются от каталогов тем, что они описывают не только действительно запомненные данные, но также и данные, которые будут запоминаться позже. Дополнительно к этому, словари данных обычно включают в себя описание формата данных. В компилированных форматах словари данных используются в системах управления базами данных, где они запоминаются на жестком диске в виде части базы данных. Кроме того, их закладывают в выходные загрузочные файлы, получающиеся в результате компиляции программы в условиях разработки программного обеспечения. Однако в патенте не заявлено представление словарей данных, в частности, пригодных для интеллектуальных карточек.

Задачей настоящего изобретения является создание средства, предназначенного для оптимизации ограничений, налагаемых ограниченными физическими размерами располагаемой области памяти в портативных устройствах обработки данных, особенно интеллектуальных карточках.

Также задачей настоящего изобретения является создание более общего механизма защищенной загрузки программных кодов и обеспечение возможности такой загрузки для множества программ, каждая из которых предназначена для одной прикладной задачи каждого портативного блока обработки данных.

Настоящее изобретение направлено также на обеспечение использования проверок режима доступа, не заданных изготовителем портативного устройства обработки данных, но выбираемых разработчиком прикладной задачи для удовлетворения его конкретных потребностей.

Для решения указанных задач соответствующая изобретению система отличается тем, что запоминающее средство дополнительно содержит по меньшей мере один контекст взаимодействия, содержащий в себе следующую структуру когерентных данных:

а) набор основных примитивов (базисных элементов) связи, которые принимаются всякий раз, когда устройство обработки данных осуществляет связь с аналогичным устройством, причем упомянутые примитивы по меньшей мере включают в себя примитив, используемый для избирательного ввода одного из упомянутых контекстов

RU 2148856 C1

RU 2148856 C1

взаимодействия;

b) набор процедурных описаний, определяющих подлежащие выполнению действия в ответ на каждый из принятых примитивов связи, по меньшей мере содержащий в себе первое процедурное описание, подлежащее выполнению при активировании контекста взаимодействия, и последнее процедурное описание, подлежащее выполнению непосредственно перед деактивированием контекста;

c) возможно незаполненный набор элементов данных, либо постоянно хранящихся в памяти, либо вычисляемых, которые полезны для использования при выполнении определяемых в процедурных описаниях процедур;

d) возможно незаполненный набор ссылок на элементы данных, и эти ссылки связываются с процедурными описаниями, причем упомянутые элементы данных также доступны для возможных в дальнейшем контекстов взаимодействия и доступны для использования при выполнении процедур, определяемых в процедурных описаниях;

e) возможно незаполненная таблица данных, содержащая в себе таблицу ссылок на элементы данных, которые доступны для точной ссылки в качестве части примитива связи, подлежащего использованию процедурным описанием, связанным с примитивом связи;

f) набор условий доступа, связанных с элементами данных, на которые осуществляются ссылки в связи с процедурными описаниями;

g) набор условий доступа, связанных с таблицей ссылок на данные в таблице данных.

Благодаря такому определению данных в запоминающем средстве портативного блока обработки блок обработки реально организован в виде процессора, то есть он не только обеспечивает возможность осуществлять логические операции, но также выполняет процедуры, которые могут загружаться в блок обработки уполномоченными на это лицами, например представителем персонала банка. Предусматривая процедуры, которые могут обеспечить произвольные сложные операции в ответ на принимаемые команды и обеспечивая подробную таблицу запомненных элементов данных, которые могут адресоваться как часть таких команд, можно оптимально использовать ширину занимаемой полосы частот, получая в результате уменьшенное количество обмениваемых команд. При соответствующей изобретению системе множество действительных использований системы требуют обмена только двумя командами. Фиксируется только структура в запоминающем средстве, которая определена таким образом, что позволяет весьма эффективно добавлять прикладные задачи блока, то есть с использованием минимально возможной дополнительной области памяти. Это особенно важно, если блок представляет собой интеллектуальную карточку, которой свойственно жесткое ограничение для располагаемой области памяти. Помимо этого, соответствующая изобретению структура обеспечивает возможности включения мер защиты для исключения доступа неуполномоченным лицам к процедурам или данным, которые они не имеют права использовать.

В первом предпочтительном варианте осуществления определенная выше система обмана данными отличается тем, что средство памяти дополнительно содержит в себе по меньшей мере два контекста взаимодействия, по меньшей мере одно описание прикладной задачи и элемент памяти для хранения ссылки на действующий в данный момент контекст взаимодействия, причем каждое описание прикладной задачи содержит в себе:

a) таблицу данных, содержащую в себе ссылки на элементы данных, эти ссылки доступны для двух или более контекстов взаимодействия и могут направляться с помощью дополнительных элементов данных;

b) дополнительный набор условий доступа, связанных с упомянутыми дополнительными элементами данных и определяющих ограничения использования.

Благодаря этим мерам все ссылки на элементы данных, которые являются общими для разных контекстов взаимодействия, доступны для всех этих контекстов взаимодействия, так что их необходимо запоминать только один раз, экономя область памяти. Кроме того, общие условия доступа к упомянутым ссылкам к данным доступны для заранее установленных контекстов взаимодействия. Следовательно, эти общие условия доступа также необходимо запоминать только один раз, экономя тем самым область памяти и повышая эффективность.

Каждое описание прикладной задачи также может содержать в себе библиотеку процедур, содержащую в себе модули рабочей программы, которые можно использовать с помощью процедурных описаний каждого контекста взаимодействия, связанного с каждым из упомянутых описаний прикладных задач.

Блок обработки предпочтительно пригоден по меньшей мере для двух прикладных задач с использованием небольшой дополнительной области памяти. Для достижения этой цели соответствующая изобретению система отличается тем, что средство памяти содержит в себе по меньшей мере два описания прикладных задач и модулей рабочей программы, которые могут быть использованы посредством процедурных описаний каждого контекста взаимодействия в каждом описании прикладной задачи или с помощью каждого модуля рабочей программы каждой библиотеки процедур в каждом описании прикладной задачи.

Модули рабочей программы в библиотеке процедур предпочтительно модернизируются путем включения спецификации использования их операционных параметров в классы, касающиеся определяющих признаков, относящихся к элементам данных, которые можно использовать в качестве действительного значения при вычислении, причем вычисление производится только в том случае, если согласованы определяющие признаки данных и классы параметров. Это является эффективным способом проверки условий доступа по информационному уровню и функциональному уровню, для

RU 2148856 C1

RU 2148856 C1

которого существует весьма эффективная реализация.

Обеспечивается более высокая надежность системы, если соответствующая изобретению система информационного обмена отличается тем, что управляющая программа содержит в себе обращение к устанавливаемому по умолчанию контексту взаимодействия, который используется для инициализирования элемента памяти, запоминающего ссылку на действующий в настоящий момент контекст взаимодействия, для выполнения конечного действия после выявления внутреннего противоречия при возвращении к нормальному режиму работы или всякий раз, когда управляющая программа активирована и не определен точный контекст взаимодействия с помощью примитива связи, принимаемого из противоположного блока обработки данных.

Для совершенствования надежности данных и функций в блоке обработки соответствующая изобретению система информационного обмена отличается тем, что средство памяти содержит контекст взаимодействия, назначенный для содержания Персональных идентифицирующих номеров, а управляющая программа приспособлена для проверки Персональных идентифицирующих номеров (ПИН), обеспечиваемых пользователем системы информационного обмена.

Контекст взаимодействия управлением Персональным идентифицирующим номером и устанавливаемый по умолчанию контекст можно выгодно реализовать в виде части той же прикладной задачи владельца устройства. Обеспечение этой прикладной задачи большим количеством устройств, с которыми осуществляет связь соответствующее изобретению устройство, дает владельцу устройства возможность пересматривать его персональные данные, запомненные в памяти устройства, например владелец интеллектуальной карточки может иметь возможность видоизменять свой ПИН (персональный идентификационный номер) на любом терминале интеллектуальных карточек, который обеспечивает соответствующий пользовательский интерфейс.

Каждое описание прикладной задачи может содержать в себе таблицу числовых значений, которая построена таким образом, чтобы обеспечить идентификаторы для всех контекстов взаимодействий, и содержит в себе по меньшей мере первое числовое значение, идентифицирующее тип прикладной задачи, второе числовое значение, указывающее уникальную идентификацию поставщика прикладной задачи, третье числовое значение, указывающее характер описания прикладной задачи, и дополнительные номера, каждый из которых однозначно указывает на один контекст взаимодействия, связанный с описанием прикладной задачи.

Последовательность числовых величин, однозначно относящихся к контексту взаимодействия, обеспечивает средство установления возможности взаимодействия между двумя осуществляющими связь устройствами, которое более эффективно, чем предусматривается в настоящее время, например, для интеллектуальных карточек, при передаче поставщику прикладной задачи ответственности за назначение однозначных значений каждому контексту взаимодействия, оставляя назначение уникальных номеров категориям и прикладным задачам для соответствующих групп пользователей, относящихся к секторной и международной кооперации соответственно. Поставщик прикладной задачи может с выгодой назначать уникальные номера контекстов для введения версий реализации и секретной информации генерирования ключа.

Средство передачи данных может быть выполнено так, чтобы осуществлять структурирование обмена данными посредством блоков данных, включающих в себя по меньшей мере две части, где первая часть представляет собой данные, квалифицируемые как рабочие данные в том смысле, что они используются для воздействия на характер операций, осуществляемых с помощью команды, как указывается примитивом связи, или для влияния на характер данных, получаемых в результате осуществления операций; вторая часть квалифицируется в качестве защиты в том смысле, что она используется для определения соответствия осуществления операции или приемлемости данных в части рабочих данных, подлежащей использованию в операции, или для подтверждения завершения операции или правильности полученных в результате данных.

Такие соответствие, приемлемость, подтверждение и правильность получаются с помощью выполнения релевантных криптографических операций с данными. Таким образом, идентификация и защита данных делаются неотъемлемой частью выполнения команд, обеспечивающих лучшую защиту, чем можно получить в имеющихся в настоящее время системах, например интеллектуальных карточках.

Управляющая программа может быть компонована для выполнения при приеме примитива связи операций, определяемых в текущем контексте взаимодействия, причем каждая операция представляет собой часть заранее установленной и фиксированной последовательности действий, каждое из которых определено как часть процедурного описания, связанного с принятым примитивом связи, и эти действия содержат в себе по меньшей мере следующие действия:

a) разрешение использования примитива связи;

b) расшифровывание рабочих данных или какой-либо их части;

c) выполнение команды с любыми входными данными;

d) зашифровывание любых рабочих данных, получающихся в результате выполнения какой-нибудь операции;

e) вычисление подтверждения завершения какого-либо выполненного действия или правильности получающихся в результате данных, подлежащих использованию при вычислениях для обеспечения защиты.

Защита дополнительно усиливается, если блок обработки данных генерирует случайное число транзакции при инициализации передачи данных, которое служит в качестве основы для криптографических вычислений.

Чтобы обеспечить возможность ввода нового контекста взаимодействия, если

RU 2148856 C1

RU 2148856 C1

требуется, один примитив связи можно назначить определенному значению, которое всегда будет интерпретироваться как требование для ввода нового контекста взаимодействия.

В следующем предпочтительном варианте осуществления соответствующая изобретению система информационного обмена отличается тем, что включает в себя дополнительный блок обработки данных, содержащий в себе такие же элементы, как блок обработки данных, а также интерфейс прикладного программирования, который состоит из кода программы, разработанного для обеспечения возможности выполнения дополнительных машинных программ для обеспечения пользователей управлением последовательностью обмениваемых примитивов связи или воздействия на передаваемые в них данные, или для обучения, или дальнейшей обработки данных, принимаемых при обмена. Совершенствование программного обеспечения для соответствующих изобретению систем обеспечивает выгоды из-за наличия интерфейса прикладного программирования.

В таком предпочтительном варианте осуществления изобретения примитив, используемый для ввода определенного контекста взаимодействия, может содержать в себе числовые значения, подлежащие использованию при вычислениях зашиты в последующих связях, первое значение, генерируемое случайным образом одним из блоков обработки, и второе значение, служащее для идентификации упомянутого блока обработки.

Дополнительные преимущества данного изобретения обеспечиваются тем, что каждый примитив связи можно далее структурировать таким образом, чтобы он состоял из двух или больше числовых значений, которые увеличивают возможности выражения значений примитива связи для интерпретации с помощью управляющей программы.

В качестве первого альтернативного варианта каждый примитив связи может состоять из двух или больше числовых значений, причем первое значение используется для обращения к процедурному описанию действия, связанного с примитивом взаимодействия, второе значение состоит из фиксированного количества двоичных значений, каждое из которых интерпретируется с помощью управляющей программы, в виде ссылки на один элемент данных.

В качестве второго альтернативного варианта каждый примитив связи может состоять из двух или больше числовых значений, причем первое значение используется для обращения к процедурному описанию действия, связанного с примитивом связи, второе значение используется для определения, который из элементов данных, доступный для внешнего обращения в действующем контексте взаимодействия, будет использован при выполнении соответствующих действий таким образом, что выбирается любой элемент данных, если он содержит значение, согласующееся с упомянутым вторым значением.

В качестве третьего альтернативного варианта каждый примитив связи состоит из двух или более числовых значений, причем первое значение используется для обращения к процедурному описанию действия, связанного с примитивом связи, второе значение составлено из ряда двоичных значений, которые присваиваются конкретным значениям управляющей программой, подлежащей использованию при интерпретировании форматов данных в примитиве связи и при осуществлении ответных действий.

Определенный выше механизм контекста и способы, которые делают его доступным, приводят к более широкому диапазону использования интеллектуальных карточек и принципу разработки прикладных задач интеллектуальных карточек, которые имеют ряд преимуществ по сравнению с традиционными путями.

Прежде всего, он обеспечивает возможность выполнения особого кода программы прикладной задачи в интеллектуальной карточке, без необходимости тщательной проверки кода на потенциальные опасности для защиты данных, запомненных для других прикладных задач. Поскольку условия доступа, которые хранятся вместе с данными на карточке, осуществляются операционной системой карточки, исключая помехи извне при выполнении кода прикладной задачи, схема мультизадачной карточки не нуждается в коде программы, проверяющем полномочия. Такие полномочия в традиционных интеллектуальных карточках являются единственным способом обеспечить возможность исполнения личного кода. При подтверждении кода для использования на карточке полномочия проверки влекут за собой обязанности в отношении защиты всей системы; это значительно усложняет управление схемами мультизадачных интеллектуальных карточек. Связанные с этим сложность и стоимость делают применение конкретного кода в традиционных схемах карточек почти невыполнимым. В случае новой технологии потребности в выполнении этих возможностей со стороны поставщиков прикладных задач интеллектуальных карточек могут быть удовлетворены.

Во-вторых, как прямое следствие защищаемой прикладной задачи специфических программ на карточках, можно реализовать специализированную прикладную задачу, которая предназначена для загрузки других прикладных задач на карточку. Таким путем прикладные задачи, загруженные на карточку, могут быть защищены от самой прикладной задачи, которая их загружает. Эта защита создает основу для деловых соглашений между сторонами, связанными с созданием схемы мультизадачной карточки, в частности между стороной, выпускающей карточки, и стороной, обеспечивающей прикладные задачи. Основываясь на материальных характеристиках, таких как величина памяти, необходимая каждой карточке, количество обеспечиваемых карточек и продолжительность выполнения прикладной задачи на карточке, вместо абстрактных понятий "траст" и "кредитное содержание", можно легче формулировать заказ поставщикам прикладных задач, чем в

RU 2148856 C1

RU 2148856 C1

традиционно выполненных интеллектуальных карточках. Более того, выпускающему карточки и поставщику прикладных задач не нужно совместно использовать ключевые слова защиты и защищать это совместное использование договорными обязательствами и взаимно согласованными возможностями переноса ключевых слов.

В-третьих, программное обеспечение прикладной задачи, если выполнено на основании новой технологии, имеет несколько преимуществ по сравнению с существующими операционными системами интеллектуальных карточек.

- Минимальный обмен данными между терминалом и карточкой необходим для установления возможности взаимодействия между карточкой и терминалом, например они поддерживают одну и ту же прикладную задачу (задачи). Значения подлежащих обмену данных для этой цели можно структурировать так, как предложено в проекте международного стандарта ISO 7816-5.

- Для завершения транзакции между карточкой и терминалом можно, как теоретически предполагается, в действительности использовать минимальное число информационных обменов, поскольку транзакция завершается в виде частного вычисления вместо необходимости использовать растянутую последовательность стандартных команд.

- Обеспечивается возможность осуществления управляемого доступа к данным, не требуя сложного пути доступа, определяемого каталогом и иерархией файлов с участием всех прикладных задач, используемых в настоящее время и предлагаемых для стандартизации.

- Обеспечивается возможность осуществления совместной разработки терминала и прикладной задачи интеллектуальной карточки, причем процесс разработки может поддерживаться компьютерными инструментальными программными средствами типа компиляторов и эмуляторов. Таким образом, разработка и реализация программного обеспечения карточки и терминала могут быть подняты над утомительным и подверженным ошибкам кодированием на общепринятом в настоящее время языке ассемблера.

- Обеспечивается возможность стандартизации оборудования, как карточек, так и терминалов с использованием абстрактного формализованного подхода для описания возможностей устройства, которые обеспечивают гибкость для будущего развития, в частности характеристик, предлагаемых изготовителями карточек и терминалов. Стандартизированные возможности терминала могут включать в себя API (интерфейс прикладных программ). В противоположность этому современные усилия по стандартизации в части интеллектуальных карточек сосредоточены на предписывании фиксированных содержаний данных сообщений, предназначенных для обеспечения идентификационных значений, интерпретируемых в соответствии со стандартом, который оставляет небольшой участок памяти для новых разработок.

И наконец, в случае новой технологии разработчики операционных систем интеллектуальных карточек имеют большую свободу в проектировании оптимальных реализаций ядра операционной системы карточки и операционной системы терминала. Разработчики аппаратного обеспечения интеллектуальных карточек имеют несколько вариантов для оптимизации использования микросхемы с поддержкой аппаратными средствами для базовых операций, включенных в ядро системы. Снижение стоимости аппаратных средств, получаемое, начиная с определенного выше специализированного проектирования, может быть больше, чем когда оно основано на усовершенствованиях универсальных однокристальных.

Теперь будет подробно описано изобретение со ссылками на чертежи, на которых представлен пример реализации общих принципов настоящего изобретения.

Фиг. 1 представляет структуру существующей прикладной программы на интеллектуальных карточках, основанную на иерархически организованной совокупности элементов данных.

Фиг. 2 представляет схему потока сообщений между портативным блоком обработки с конструированным аналогичным образом блоком обработки в принятом в настоящее время в качестве стандарта формате.

Фиг. 3 представляет основную реализацию настоящего изобретения с использованием концепции контекстов взаимодействия в портативных блоках обработки, таких как интеллектуальные карточки и терминалы карточек.

Фиг. 4 представляет пример практической организации управляющего контекста, выделяющей различные взаимосвязи между процедурными описаниями, содержащимися в контексте взаимодействия и элементах данных, и библиотечными функциями, используемыми при выполнении процедур.

Фиг. 5 представляет пример блок-схемы управления выполнением программы и переключений контекстов для обеспечения защиты, связанных с выполнением процедурного описания, активизируемого примитивом связи.

На фиг. 1 представлена структура данных и файлов в существующих системах. По существу, имеется главный файл 1, который подсоединен к нескольким элементарным файлам 3 и одному или больше специализированным файлам 2. Каждый специализированный файл 2 может быть подсоединен к одному или больше следующим специализированным файлам 2 и к одному или больше элементарным файлам 3. В известном уровне техники используют древовидную иерархию каталогов и файлов. Количество подчиненных уровней в известной структуре в принципе не ограничено. Используемая в фиг. 1 терминология взята из предложенного международного стандарта ISO 7816-4. В соответствии со стандартом формата коммуникационного потока между портативным блоком обработки данных 5 и аналогично структурированным блоком обработки данных 4, как показано на фиг. 2, осуществление связи включает в себя набор блоков. Осуществление связи начинается с сигнала установки m0 из блока обработки

RU 2148856 C1

RU 2148856 C1

данных 4. Такой сигнал установки может быть вне занимаемой ширины полосы частот типа генерируемой посредством логической схемы включения питания в блоке обработки данных 5. Портативный блок обработки данных 5 отвечает сигналом ответа на сигнал установки (ATR) m1 по возможности с последующими содержаниями. Все последующие пары блоков m2, m3,..., m(n-1), mn состоят из блоков, возглавляемых примитивом связи (например, командой), за которым следует содержание.

На фиг. 3 показана внутренняя структура двух соответствующих изобретению блоков обработки данных, которые обмениваются друг с другом передаваемыми и принимаемыми данными. Расположенный слева блок обработки данных 4 может представлять собой терминал, а расположенный справа блок обработки - портативный блок обработки данных, например интеллектуальную карточку. Однако изобретение также применимо к двум портативным блокам обработки данных, способным осуществлять связь друг с другом с помощью соответствующего средства связи.

Каждый из блоков обработки данных 4, 5 содержит в себе средство 7, 14 передачи данных, посредством которого можно обмениваться структурированными блоками данных. Каждый из блоков обработки данных 4, 5 содержит в себе средство обработки 8, 15 и память 9, 16. Память 9, 16 может иметь любую конфигурацию постоянного запоминающего устройства (ПЗУ), запоминающего устройства с произвольной выборкой (ЗУПВ) и программируемого постоянного запоминающего устройства, типа электрически стираемого программируемого постоянного запоминающего устройства (ЭСППЗУ).

Память 9, 16 включает в себя управляющую программу 12, 17, на фиг. 3 обозначенную "MAXOS". Если портативный блок обработки данных 5 пригоден для двух или больше прикладных задач, память 9, 16 содержит в себе два или больше описаний прикладных задач 13(1)... 13(n), 18(1)...18(n). Здесь имеется столько описаний прикладных задач, сколько прикладных задач содержит рассматриваемый блок обработки данных. Каждое описание прикладной задачи обозначено ссылочной позицией CSA. Второе описание прикладной задачи 13(2), 18(2) показано в увеличенном масштабе на фиг. 3 для отображения содержимого каждого описания прикладной задачи. Каждое описание прикладной задачи 13(i), 18(i) содержит в себе по меньшей мере один "контекст взаимодействия" 11(1).. . 11(m), 19(1). ..19(m). Каждый контекст взаимодействия обозначен ссылочной позицией CTA. Первый из этих контекстов взаимодействия 11(1), 19(1) показан в увеличенном масштабе для отображения его содержимого. Каждый контекст взаимодействия содержит в себе:

- набор команд, определяющих примитивы связи, распознаваемые контекстом взаимодействия и указывающие соответствующие процедуры, определяемые в наборе процедур;
- набор данных;
- набор ссылок на данные, постоянно хранящиеся в других контекстах взаимодействия, если вообще имеются;
- набор процедур, которые могут выполняться с помощью управляющей программы 12, 17;
- набор условий доступа к элементам данных;
- набор внешних ссылок, указывающих на элементы данных, подлежащие использованию в командах, выдаваемых другим блоком обработки данных;
- дополнительно, другие перечни, определенные разработчиком.

Наконец, память 9, 16 содержит элемент памяти 21, 20, который содержит ссылку на "текущий CTA", то есть на действующий в данный момент контекст взаимодействия.

Назначение различных контекстов взаимодействия внутри одного описания прикладной задачи состоит в том, чтобы обеспечить функциональное разделение в возможных взаимодействиях между блоками обработки данных 4, 5. Это особенно релевантно, когда функциональное разделение также является разделением условий защиты. Примером может служить первое взаимодействие между интеллектуальной карточкой и терминалом с целью, например, открывания двери и второе взаимодействие для программирования дверей, которые можно открыть. Второе взаимодействие нуждается в лучшей защите, чем первое взаимодействие, и для него назначается собственный контекст взаимодействия. Для получения доступа к контексту взаимодействия первый этап заключается в обеспечении защиты операций, которые могут выполняться в пределах контекста взаимодействия.

На фиг. 4 показан практический подход к реализации механизма контекста, отображенный в виде модели организации памяти, которая изображает соотношения между элементами данных, условиями доступа и процедурами. Структура по фиг. 4 применяется тогда, когда имеются две или больше прикладные задачи для портативного блока обработки данных 5. Если имеется только одна прикладная задача, структура сильно упрощается, как будет показано ниже. На фиг. 4 изображены ссылочные позиции блока обработки данных 5. Однако показанная на фиг. 4 структура также применима к памяти 9 блока обработки данных 4. На фиг. 4 описания элементов данных и описания процедур оптимально организованы таким образом, чтобы отражать разделение кода программы и разделение данных между разными контекстами взаимодействия (ссылочные позиции CTA), которые составляют одну прикладную задачу (CSA).

Память 16 содержит элементы данных H(1)...H(7), элементы выполняемого кода G(1). ..G(5), которые являются частью операционной системы, и описания прикладных задач 18(1), 18(2) (CSA1, CSA2). На фиг. 4 данные и код, которые являются внутренними по отношению к операционной системе, вынесены влево. Количество элементов данных, элементов исполняемого кода и описаний прикладных задач, представленное на фиг. 4, дано только в качестве примера. В действительности количества могут изменяться в зависимости

RU 2148856 C1

RU 2148856 C1

от потребностей.

Каждое описание прикладной задачи 18(1), 18(2) физически присутствует в памяти. Они обеспечивают первый нижний уровень абстракции для отражения использования памяти. Каждое описание прикладной задачи 18(1), 18(2) состоит из:

- библиотеки процедур, состоящей из блоков исполняемого кода F(1)... F(4), которые могут указывать на блоки исполняемого кода операционной системы, сделанные пригодными для этой цели, как показано стрелками p(1)...p(5);

- таблицы элементов данных E(1)...E(7), подлежащих использованию с помощью процедур в контекстах взаимодействия 19(1)...19(2) в настоящем описании прикладной задачи 18. Эта таблица данных содержит в себе условия доступа к данным и указатели q(1)...q(7) на области памяти, хранящие элементы данных;

- таблицы контекстов взаимодействия, содержащей в себе ряд описаний контекстов взаимодействия 19(1), 19(2).

Количество элементов в библиотеке процедур, количество элементов данных и таблица контекстов взаимодействий описания прикладной задачи 18(1), как показано на фиг. 4, предназначены только для целей представления. Конечно, количество элементов может меняться в зависимости от требований прикладной задачи.

Контексты взаимодействия 19(1), 19(2) физически находятся в запоминающем средстве, хранящем описание прикладной задачи 18(1). Логически контексты взаимодействия обеспечивают второй уровень управления использованием памяти. Объединенное управление, обеспечиваемое этим вторым уровнем и уровнем описания прикладной задачи, дает эффективную реализацию механизма контекстов выполнения, предназначенного для портативных блоков обработки данных типа интеллектуальных карточек. Каждый контекст взаимодействия 19(1), 19(2) содержит в себе

- таблицу процедурных описаний C(1)...C(5). Эти процедурные описания могут указывать на процедурные описания в процедурной библиотеке в описании прикладной задачи 18, как показано для примера стрелками s(1), s(2). В качестве альтернативы эти процедурные описания могут указывать на элементы исполняемого кода G(1)...G(5), обеспечиваемые операционной системой, как показано, например, стрелкой t(l). В качестве еще одной альтернативы эти процедурные описания могут содержать подобные ссылки на какие-либо элементы данных, которые используются процедурой во время ее выполнения и которые присутствуют в таблице данных рассматриваемого описания прикладной задачи 18, как показано стрелками r(1)...r(6);

- таблицу данных, содержащую элементы данных B(1)...B(5), исключительно пригодных для использования процедурами в рассматриваемом контексте взаимодействия. Элементы данных представлены как ссылки на таблицу данных рассматриваемого описания прикладной задачи 18 вместе со связанными с ними условиями доступа, придерживаясь этого при обращении к действительным данным, как показано стрелками u(1)...u(5);

- таблицу внешнего интерфейса, содержащую примитивы связи A(1)...A(4), которые принимаются в качестве команд рассматриваемыми контекстами взаимодействия 19(1), 19(2). Каждая команда в примитиве связи указывает на элемент процедурных описания C(1)...C(5) таблицы процедур в рассматриваемом контексте взаимодействия, как показано стрелками v(1)...v(4). Команды, поступающие из устройства установления связи 4, могут указывать на элементы в таблице данных описания прикладной задачи с помочью одного адреса или больше, следующих за командой. Каждая команда может сопровождаться элементами данных в качестве входного сообщения для обработки команды. Количество адресов, как это приведено здесь, дано лишь в качестве примера и определяется для каждой команды, как это требуется в действительности.

Защита элементов данных обеспечена условиями доступа. Любая внешняя команда в примитиве связи A(1)...A(4) может адресоваться только к элементам данных, упоминаемым в таблице данных рассматриваемого контекста взаимодействия 19. Доступ оказывается возможным только в том случае, если удовлетворяются условия доступа. Эти условия доступа определяют тип доступа, который допускается для команды; таким условием доступа может быть отсутствие доступа, доступ только для чтения, доступ для чтения и записи и использование секретного ключевого слова. Можно также применять другие условия доступа. Например, команда примитива связи A(1) может иметь доступ только для чтения к элементу данных B(2) посредством обращения по стрелке w(2), тогда как команда примитива связи A(2) имеет доступ для чтения и записи к тому же элементу данных B(2) посредством обращения по стрелке w(3).

Процедурные описания C(1). ..C(5) могут указывать на элементы данных в таблице данных рассматриваемого описания прикладной задачи 18, но не других. И здесь доступ обеспечивается только в том случае, если удовлетворяется условие доступа. Эти условия доступа, кроме того, определяют тип доступа, в качестве которого может быть, например, отсутствие доступа, доступ только для чтения, доступ для чтения и записи и использование секретного ключевого слова. Условия доступа для различных процедурных описаний в одном и том же контексте взаимодействия 19 могут различаться для одного и того же элемента E(1)...E(7) таблицы данных описания прикладной задачи, например стрелка r(1) может представлять условия доступа только для чтения, тогда как стрелка r(2) может представлять условие доступа для чтения и записи.

Условия доступа проверяются на релевантном уровне, то есть уровне описания прикладной задачи или уровне контекста взаимодействия и только однажды. Элемент B(1)...B(5) таблицы данных в контексте взаимодействия 19(1), 19(2) указывает непосредственно стрелкой u(1)...u(5) на указатель элемента данных в таблице данных описания прикладной задачи 18(1), поскольку

RU 2148856 C1

RU 2148856 C1

условия доступа уже удовлетворяются в элементе E(1)...E(7) таблицы данных описания прикладной задачи 18(1). Однако процедурные описания C(1)...C(5) в контексте взаимодействия 19(1), 19(2), которые указывают на элементы таблицы данных в описании прикладной задачи 18(1), должны вначале удовлетворять условию доступа, связанному с элементами E(1). . .E(7) таблицы данных в описании прикладной задачи 18(1).

Ни на какие элементы данных или элементы процедурного описания в таблицах данных описания прикладной задачи 18(1) и связанные с ними контексты взаимодействия 19(1), 19(2) не может указывать никакое другое описание прикладной задачи в памяти 16. Исполняемый код, который составляет процедурное описание, может адресовать данные только косвенно, посредством ограниченного набора информационных ссылок, связанных с каждым из процедурных описаний C(1)...C(5). Используя элементы данных, описываемые посредством B(1). . .B(5), таблица ссылок временно расширяется с помощью управляющей программы ссылками на элемент данных, получаемых путем оценки адресов, которые в действительности определяются в коммуникационном сообщении, принятом в качестве команды, связанной с процедурным описанием.

Таким образом, нельзя осуществлять доступ к данным, кроме точно определенных и только при соблюдении определенных условий использования. Другими словами, показанная на фиг. 4 предпочтительная эталонная модель ссылки на память в отношении описания прикладной задачи вместе со связанными с ним контекстами взаимодействия обеспечивает эксклюзивный контекст, предназначенный для операций в пределах одной единственной прикладной задачи блока обработки данных 5. Элементы данных H(1)...H(7) хранятся в памяти 16, общей для всех прикладных задач, но содержат данные, предназначенные для исключительного использования в контексте описания прикладной задачи 18(1), такая исключительность гарантируется выполняемой программой путем допущения существования единственного указателя для каждой ячейки памяти типа q(1) от E(1) до H(2). Только на элементы кода G(1)...G(5) может указывать какое-либо из описаний прикладной задачи 18(1). . ., хранящихся в памяти 16. Эти последние ссылки иного описания прикладной задачи, чем описание прикладной задачи 18(1) на общие коды G(1)...G(5), не показаны явно на фиг. 4. Однако любой специалист в данной области техники легко может расширить показанную на фиг. 4 структуру до двух или более описаний прикладных задач 18(1), 18(2),...

После объяснения того, какие элементы данных можно защитить с помощью использования условий доступа различного рода, теперь будет объяснено обеспечение управления памятью. Для управления управляющим устройством желательно, чтобы изменяемые данные (элементы данных) и неизменяемые данные (код операционной системы) могли управляться операционной системой раздельно. Как показано на фиг. 4, эталонная модель памяти обеспечивает разделение кода и элементов данных в памяти 16, на которые указывают указатели q(1)...q(7), p(1)...

p(5) из таблицы данных и библиотеки процедур соответственно в рассматриваемом описании прикладной задачи 18. Элементы таблицы данных в каждом контексте взаимодействия 19(1), 19(2) содержат только ссылки на эти указатели и не указывают непосредственно на коды G(1)...G(5) и элементы данных H(1)...H(7) в памяти 16. Таблица данных рассматриваемого описания прикладной задачи 18 обеспечивает уровень использования косвенной адресации, требуемый операционной системой для осуществления управления памятью.

Дублирование кода предотвращается путем обеспечения общих библиотек кодов на двух уровнях: "тела команд", подобные процедурному описанию C(3), которое указывает на элемент кода F(2) в библиотеке процедур в описании прикладной задачи 18(1) для распределения общих кодов между различными контекстами взаимодействия. Однако тело процедурного описания C(3) указывает также непосредственно на код G(3), хранящийся в памяти 16 и обеспеченный операционной системой. Все блоки исполняемого кода G(1)...G(5), обеспеченные операционной системой, введены для эффективного выполнения.

По своей сути соответствующая фиг. 4 структура памяти применима также в ситуациях, где обеспечена только одна прикладная задача блока обработки данных 5. В этом случае только описание прикладной задачи 18(1) может четко совпадать с одним контекстом взаимодействия 19(1), и этот контекст взаимодействия в таком случае содержит в себе по меньшей мере следующую структуру когерентных данных:

a) набор основных примитивов связи A(1)..., которые принимаются всякий раз, когда блок обработки данных 5 связывается с подобным блоком 4, причем упомянутые примитивы по меньшей мере включают в себя примитив, используемый для избирательного ввода одного из упомянутого по меньшей мере одного из контекстов взаимодействия;

b) набор процедурных описаний C(1)..., определяющих действия, подлежащие выполнению в ответ на каждый из принимаемых примитивов связи A(1)..., по меньшей мере содержащий в себе первое процедурное описание, подлежащее выполнению при активировании контекста взаимодействия, и последнее процедурное описание, подлежащее выполнению непосредственно перед деактивированием контекста;

c) возможно незаполненный набор элементов данных H(1)..., либо постоянно хранящихся в памяти, либо вычисляемых, которые доступны для использования, когда выполняются процедуры, определяемые в процедурном описании C(1)...;

d) возможно незаполненный набор ссылок на элементы данных, причем эти ссылки связаны с процедурными описаниями C(1)..., а упомянутые элементы данных доступны также для возможных дополнительных контекстов взаимодействия и доступны для использования, когда выполняются процедуры, определяемые в процедурных описаниях C(1)...;

RU 2148856 C1

RU 2148856 C1

e) возможно незаполненная таблица данных, содержащая в себе таблицу ссылок на элементы данных, которая пригодна для точной ссылки в качестве части примитива связи, подлежащего использованию процедурным описанием, связанным с примитивом связи;

f) набор условий доступа, связанный с элементами данных, которые указываются в связи с процедурными описаниями;

g) набор условий доступа, связанный с таблицей информационных ссылок B(1)... в таблице данных.

Если для блока обработки данных 5 обеспечена только одна прикладная задача и имеется по меньшей мере два контекста взаимодействия 19(1), 19(2), то каждое описание прикладной задачи содержит в себе:

a) таблицу данных, содержащую указания E(1)... на элементы данных, причем эти ссылки могут быть доступными для двух или более контекстов взаимодействия 19(1)... и могут быть расширены дополнительными элементами данных;

b) дополнительный набор условий доступа, связанный с упомянутыми ссылками E(1). .. или с упомянутыми дополнительными элементами данных и определяющий ограничения использования.

Набор процедурных описаний в каждом из двух или больше описаний контекстов взаимодействия также содержит дополнительное последнее процедурное описание, подлежащее выполнению непосредственно перед деактивированием контекста.

Фиг. 5 представляет поток управления в управляющей программе, определяемой выше символом "MAXOS" (12, 17).

После включения питания системы программное обеспечение начинает обработку кода установки на этапе 30. На этапе 31 вводят уровень защиты ядра операций блока обработки данных. Описывающие этот уровень условия доступа хранятся в неизменяемой части памяти, например в ПЗУ или в логических схемах аппаратного средства. На этапе 32 осуществляется проверка энергонезависимого запоминающего устройства на совместимость, и любые модификации, которые могли остаться незаконченными из-за внезапного отключения энергии, например из-за извлечения интеллектуальной карточки, аннулируются. Проверка на непротиворечивость энергонезависимого запоминающего устройства включает в себя только исследование информации о состоянии, хранящейся в памяти, и вычисление контрольных сумм. Содержимое запоминающего устройства, если вообще обеспечивается доступ, используется только для вычисления контрольных сумм. Таким образом, проверка на непротиворечивость представляет безопасную операцию. Точный характер возможностей проверки на непротиворочивость зависит от деталей аппаратного обеспечения в блоке обработки данных и программ модификации энергонезависимого запоминающего устройства, которые в значительной степени не соответствуют определенной структуре защиты. После общей проверки совместимости памяти проверяются предварительно рассчитанные уровни контекста защиты, хранящиеся в памяти. Наконец, инициируется память с произвольной выборкой блока обработки данных.

На этапе 33, если условие выполнения программы таким образом объявлено надежным, вводится надежный уровень защиты прикладной задачи блока обработки данных. На этом уровне блокируется любой доступ в память, имеющий отношение к операциям ядра. Доступ к данным и описанию прикладной задачи с этого уровня обеспечивается исключительно через подпрограммы в ядре, которые поддерживают информацию о состоянии в текущих операциях памяти.

При первом вводе после установки на этапе 34 дескрипторы элементов данных прикладной задачи используются для проверки совместимости запомненных данных с дескриптором, и в память вносится изменение, если они находятся в несовместимом состоянии с определяющим признаком, как описывалось. Сообщение ответа на установку формируется из идентификаторов прикладной задачи, хранящихся в описаниях прикладной задачи, и завершается номером транзакции, вычисленным как непрогнозируемое при приеме другим блоком обработки данных 4. Для активирования устанавливаемого по умолчанию контекста взаимодействия вырабатывается внутренняя в отношении блока обработки данных команда терминала. Непосредственно после посылки сообщения ответа на установку на другой блок обработки данных 4 выполняется эта внутренняя команда активирования контекста с целью обеспечения контекста взаимодействия, предназначенного для последующих команд. Сообщение ответа на установку ясно показывает готовность блока обработки данных 5 к приему следующих команд. Устанавливаемый по умолчанию контекст взаимодействия может быть создан в виде части "прикладной задачи владельца интеллектуальной карточки", которая присутствует в качестве одной стандартной прикладной задачи во всех мультизадачных интеллектуальных карточках. В этом специфическом контексте прикладной задачи пользователь, то есть владелец интеллектуальной карточки, может пересматривать свои личные данные или открыть на карточке любую из других прикладных задач.

На этапе 35 в результате команды активирования контекста вводится уровень защиты контекста взаимодействия (CTA) для стандартного CTA владельца интеллектуальной карточки.

После полного активирования прикладной задачи она оказывается готовой принимать команды с другого блока обработки данных 4. Дальнейшая обработка зависит от принятой команды; управление командой активирования прикладной задачи отличается от управления командой, которая подлежит исполнению. Следовательно, на этапе 38 после установления, что примитив связи принят на этапе 36, и установления его приемлемости на этапе 37 проводится тестирование, нужно ли активировать новую

RU 2148856 C1

прикладную задачу. Если нет, то осуществляется переход к этапу 39, на котором команда проверяется с целью определения, допустима ли она и могут ли быть приняты входные данные. Эти проверки команды осуществляются только в том случае, если они определены в дескрипторе прикладной задачи. Кроме того, на этапе 39 можно выполнить дешифровку входных данных.

Если тестирование успешно, то на этапе 40 вводится "уровень защиты доступа к данным". На этом уровне, самом высоком уровне защиты, могут выполняться программы (этап 41), которые закодированы поставщиком прикладных задач. Такие программы хранятся в дескрипторе прикладных задач и функционируют как специальная реакция прикладной задачи на конкретную команду, выдаваемую другим блоком обработки данных 4. Этот уровень защиты ограничивает доступ в память для подмножества, специально определенного для выполняемой команды.

После выполнения на этапе 41 команды с представленными входными данными уровень защиты доступа к данным оставляется (этап 42).

На этапе 43 вырабатывается завершение формирования выходных данных и (криптографическая) проверка выполнения команды. После этапа 43 программа ожидает на этапе 36 новые примитивы связи.

Если не определена программа специальных команд и команда может выполняться с помощью процедур, состоящих исключительно из функций операционной системы, уровень защиты доступа к данным (этап 40) не вводится, и команда будет выполняться непосредственно на уровне защиты контекста взаимодействия, поскольку программы операционной системы разрабатывают без нарушения какой-либо защиты данных.

Если на этапе 38 устанавливается, что новой подлежащей активированию прикладной задачи нет, программа переходит к этапу 44, на котором осуществляется процесс деактивирования контекста. На этапе 45 оставляется специальный уровень защиты текущей прикладной задачи и на этапе 46, в пределах уровня защиты исполняемой программы "MAXOS", проверяются данные, сопровождающие команду.

Если команду можно идентифицировать в качестве определяемой для запрашиваемой прикладной задачи, на этапе 47 вводится новый уровень защиты специального СТА (контекста) прикладной задачи. Этот уровень ограничивает доступ к данным, имеющим отношение к вновь открытой прикладной задаче.

Блок обработки данных 5 вырабатывает данные в ответ на команду активирования контекста посредством выполнения на этапе 48 команды инициализации, определяемой в таблице процедур. Если присутствует такая закодированная программа поставщика прикладных задач, то на этапе 49 вводится уровень защиты доступа к данным. На этапе 50 выполняется процесс активирования контекста. На этапе 51 уровень защиты доступа к данным оставляется, и передается ответ на другой блок обработки данных 4, а сам блок обработки данных 4 готов принимать новую команду после описанного выше этапа 43.

После описания фиг. 1 - 5 будут представлены некоторые общие комментарии к соответствующей изобретению системе информационного обмена.

Коды в библиотеке процедур в каждом описании прикладных задач 18(1), 18(2) можно модифицировать посредством включения описания использования их операционных параметров в классы, относящиеся к определяющим признакам, имеющим отношение к элементам данных, которые можно прогонять в качестве действительных значений при вычислении, причем вычисление происходит только тогда, когда согласуются определяющие признаки данных и классы параметров. Это обеспечивает возможный путь проверки условий доступа как к элементам данных, так и к функциям. Сравнивая надлежащим образом закодированные битовые отображения определяющих признаков данных с классами параметров соответственно можно обеспечить эффективную реализацию этого дополнительного технического приема.

Управляющая программа 12, 17 может содержать в себе ссылку на контекст взаимодействия, который используется для инициализации текущего контекста взаимодействия в элементе памяти 20, хранящем ссылку на действующий в данный момент контекст взаимодействия. Благодаря такой мере можно выполнить окончательное действие после выявления внутренней несовместимости при возвращении операции в нормальное состояние или всякий раз, когда управляющая программа 12, 17 активизирована, а точный контекст взаимодействия не определен примитивом связи, полученным от другого блока обработки данных 5. Этот устанавливаемый по умолчанию контекст взаимодействия может быть таким контекстом, который содержится в прикладной задаче владельца карточки, как описывалось выше.

Дополнительно к этому, память 9, 16 может содержать в себе контекст взаимодействия 11, 19, предназначенный для включения персональных идентифицирующих номеров (ПИН) и исполнительную программу 12, 17, приспособленную для проверки персональных идентифицирующих номеров, обеспечиваемых пользователем системы информационного обмена. Можно использовать несколько таких персональных идентифицирующих номеров, паролей. Один такой пароль можно использовать для защиты использования устройства при транзакциях, где можно обнаружить уязвимые данные секретности. Второй пароль можно использовать для защиты транзакций, когда передаются данные, представляющие значение, оплачиваемое владельцем пароля. Третий пароль можно использовать для защиты транзакций, где выполняются операции, считающиеся критическими для защиты прикладной задачи, типа режимов защиты, считающейся необходимой вызову, как это установлено, в каждом из контекстов взаимодействия 11, 19, которые могут это потребовать. Могут быть обеспечены дополнительные кодовые слова. Этот контекст взаимодействия организации ПИН

RU 2148856 C1

RU 2148856 C1

может быть таким контекстом, который содержится в прикладной задаче владельца карточки, как описано выше.

Каждое описание прикладной задачи 13, 18 может содержать в себе таблицу числовых значений, которая составлена для обеспечения идентификаторов, предназначенных для всех контекстов взаимодействия 11, 19, и каждое описание прикладной задачи 13, 18 может содержать в себе по меньшей мере первое числовое значение, показывающее тип прикладной задачи, второе числовое значение, показывающее уникальную идентификацию поставщика прикладной задачи, третье числовое значение, показывающее характер описания прикладной задачи 13, 18, и следующие числа, каждое из которых однозначно указывает на один контекст взаимодействия 11, 19. Первые два числа могут быть назначены в соответствии с правилами, установленными в торговле, тогда как оставшиеся числа могут выбираться поставщиком прикладной задачи. В частности, числовые значения можно назначать для обозначения отличительных признаков между разными вариантами выполнения или идентификации формирования группы криптографических ключей, используемых прикладной задачей при ее криптографических вычислениях. Кроме того, устройство может включать в ответе на сообщение установки таблицу, предназначенную для каждого из контекстов взаимодействия 11, 19, содержащихся в памяти, идентификационное число, составленное из уникальных идентификационных значений, хранящихся вместе с контекстом взаимодействия. Первый элемент в таблице идентификационных чисел контекста взаимодействия может представлять идентификацию устанавливаемого по умолчанию контекста.

Средство передачи данных 7, 14 предпочтительно выполнено для обмена структурированными данными в виде блоков данных. Эти блоки данных содержат по меньшей мере две части, из которых первая часть представляет собой данные, определяемые как операционные, поскольку они используются для воздействия на характер операций, выполняемых по команде, как показано примитивом связи, или данными, получаемыми при выполнении операций. Вторая часть квалифицируется в качестве защиты в том отношении, что она используется для определения соответствия выполнения операции или приемлемости данных в операционной части, используемых в операции, или для проверки завершения операции, или правильности полученных данных.

Когда данные структурированы таким образом, управляющую программу 17 можно приспособить для выполнения, при приеме примитива связи, операций, определяемых в текущем контексте взаимодействия 20, 21, каждой операции как части заранее установленной и постоянной последовательности действий, каждое из которых определено отдельно в виде части правила описания процедур, связанного с принимаемым примитивом связи. Первое действие можно определить в виде функции, предназначенной для санкционирования использования примитива связи а этой точке в последовательности передач данных. Второе действие можно определить как функцию, предназначенную для дешифровки операционных данных или какой-либо их части, тогда как третье действие можно определить как собственно операционную процедуру. Четвертую часть можно определить для шифрования каких-либо операционных данных, получаемых в результате выполняемых операций, а пятое действие можно определить в качестве функции для вычисления проверки завершения выполнения действия или правильности полученных в результате данных, или для использования при вычислениях защиты в принимающем блоке обработки данных. Эти действия отображены в блок-схеме по фиг. 5.

Кроме того, блок обработки данных 5 может включать в своем сообщении ответа на установку число, выбранное таким, чтобы оно было непредсказуемо по значению для принимающего блока обработки данных 4, которое может служить в качестве основы для криптографических вычислений. Такое число можно обозначить как "номер транзакции карточки".

Здесь будет обеспечено для одного назначенного примитива связи определенное значение, которое всегда будет интерпретироваться как запрос на ввод нового контекста взаимодействия 11, 19. Этот примитив связи можно обозначить "командой активизации". Данные, сопровождающие команду активизации, в достаточной мере определяют подлежащий активированию контекст, возможно, путем ссылки на идентификационные числа, передаваемые в виде части ответа на сообщение о восстановлении. Действия, выполняемые в ответ на команду активирования, во-первых, описываются с помощью процедурного описания, содержащегося в контексте, принимающем обозначенный примитив, в отношении деактивирования, и, во-вторых, описаны в процедурном описании, предназначенном для активирования, содержащемся в контексте, определяемом как подлежащий вводу.

Примитив связи, используемый для ввода определенного контекста взаимодействия 11, 19, предпочтительно содержит числовые значения, подлежащие использованию в вычислениях защиты при последующих передачах данных. Первое произвольное значение может вырабатываться с помощью одного из блоков обработки 4, 5, а второе значение может служить для идентификации этого одного блока обработки. Эта идентификация может быть результатом вычислений, которые выполняются так, что получающееся в результате значение в достаточной мере идентифицирует устройство и состояние его памяти, как это требуется вычислениями или другими действиями, которые могут осуществляться при последующих обменах данными в контексте взаимодействия 11, 19, подлежащем активированию. Упомянутое второе значение можно обозначить "идентификацией терминала".

Кроме того, команда активирования дается в качестве части полученных в результате данных, - числовое значение,

RU 2148856 C1

RU 2148856 C1

служащее для идентификации в достаточной степени конкретного отвечающего блока обработки данных, как этого требуют вычисления или другие действия, которые могут проводиться при последующих обменах данными в только что активированном контексте, и это число можно обозначить "идентификацией интеллектуальной карточки".

Помимо этого, идентификационный номер интеллектуальной карточки можно вычислить, используя криптографические функции из данных, хранящихся в блоке обработки данных 5, или из данных, принимаемых в виде части команды активирования, таким образом, что номер меняется непредсказуемым образом при вычислении в ответ на команды активирования, принимаемые с запускающих устройств с разными идентификационными номерами оконечного устройства, таким образом, рассчитанную идентификацию интеллектуальной карточки можно обозначить "псевдонимом интеллектуальной карточки". Более того, до выполнения действий, описанных в процедурном описании процедуры активирования подлежащего вводу контекста, управляющая программа может выполнять криптографическое вычисление, определяемое в виде части процедурного описания в данном обозначенном контексте, подлежащем выполнению при активировании с целью определения, может ли быть активирован контекст. Вычисления могут включать в себя использование идентификации транзакции интеллектуальной карточки, идентификацию транзакции терминала и идентификацию терминала и другие значения, хранящиеся в памяти.

В качестве альтернативы таким конкретным вычислениям, поддерживаемым конкретными данными при выполнении команд, можно использовать команды с описанием битового поля упоминаемых элементов данных. Затем, каждый примитив связи составляется из двух или больше числовых значений, где первое значение используется для указания на процедурное описание действия, связанного с примитивом связи, второе значение состоит из фиксированного количества двоичных значений, каждое из которых интерпретируется управляющей программой 12, 17 в качестве ссылки на единственный элемент данных. Этот элемент данных определен в таблице ссылок на внешние данные в рассматриваемом контексте взаимодействия 11, 19, причем каждый элемент данных в таблице определен наличием двоичного значения одного из двоичных чисел в соответственной позиции в таблице двоичных значений. Это второе значение можно обозначить "адресом операнда". Каждый из определенных таким образом элементов данных делается пригодным с помощью действующей управляющей программы 12, 17, подлежащей использованию в ответном действии, таким образом, который можно описать в процедурном описании этого действия.

В качестве альтернативы конкретным вычислениям с конкретными данными и командами с описанием битового поля эталонных элементов данных можно использовать формат команды с описанием согласования данных элементов данных. В этом случае каждый примитив связи состоит из двух или больше числовых значений, где первое значение используется для указания на процедурное описание действия, связанного с примитивом связи, второе значение используется для определения, которое из доступных элементов данных будет использовано для внешней ссылки в активном контексте взаимодействия 12, 19 при выполнении ответных действий таким образом, что выбирается какой-нибудь элемент данных, если он содержит значение, которое соответствует упомянутому второму значению. Это второе значение можно обозначить "определителем признака операнда". Дополнительно к этому, контекст взаимодействия 11, 19 может содержать процедурное описание, показывающее, каким образом определитель признака операнда, заданный в виде части команды, должен сравниваться с данными, содержащимися в каком-нибудь из элементов данных, доступных для внешней ссылки в данном контексте, и это процедурное описание выполняется с целью выбора предполагаемых элементов данных до выполнения процедурного описания, определяющего надлежащие действия команды.

В качестве еще одной альтернативы можно использовать формат команды со спецификацией битового поля интерпретации команды. В этом случае каждый примитив связи состоит из двух или больше числовых значений, где первое значение используется для обращения к процедурному описанию действия, связанного с примитивом связи, второе значение состоит из ряда двоичных значений, которые назначаются конкретному значению управляющей программой 12, 17, подлежащему использованию в интерпретирующих формах данных в примитиве связи и при выполнении ответных действий. Здесь второе значение можно обозначить "модификатором команд". Значения распознаются для их назначенных значений всеми блоками, обеспеченными этим дополнительным методом.

В случае применения последней альтернативы модификатор команд может включать в себя двоичное значение, которое определяет, подлежит ли использованию третья часть команды в качестве адреса операнда или в качестве определителя признака операнда. Однако модификатор команды в качестве альтернативы может включать в себя двоичное значение, которое определяет, использует ли операция, выполняемая в качестве ответа на команду, данные в виде одного элемента данных, или состоит ли из конкатенации элементов данных, подлежащих обработке в связи с каждым элементом данных, определенным как часть значения команды, используя адреса операнд, или спецификатор признаков операнд. В качестве альтернативы модификатор команды может включать в себя двоичное значение, которое определяет, закодированы ли данные, обеспеченные командой, с использованием метода признак-длина-значение, для отличия последующих соединенных элементов данных.

RU 2148856 C1

RU 2148856 C1

Следующий вариант заключается в том, что модификатор команд может включать в себя двоичное значение, которое определяет, действительно ли осуществление действия, выполняемое по команде, ведет к эффективному изменению данных, хранящихся в блоке обработки данных 5 (интеллектуальной карточке), или на самом деле дает данные, рассчитываемые блоком обработки данных 5, или что результатом этой команды являются данные, отражающие состояние блока относительно приемлемости команды, сопровождающих ее данных, объема данных, которые можно получить в результате расчетов, или других различных определяющих признаков.

Короче говоря, представленный выше новый способ особенно пригодный для внедрения в интеллектуальные карточки, является концепцией отдельного условия выполнения программы. В этом способе средство обработки и другие ресурсы в вычислительной машине распределяются между различными прикладными задачами, как если бы прикладная задача была только пользователем вычислительной машины. Для определения условий множественного доступа для данных, совместно используемых рядом связанных прикладных задач, обеспечена реализация этого нового способа в интеллектуальных карточках в дополнение к механизму обработки информации. Второй способ, обеспечиваемый отдельными условиями выполнения программы и приведенный выше, представляет собой возможность определения функционального смысла команд в каждом условии для получения минимального количества команд в каждом взаимодействии между двумя подобными блоками обработки данных 4, 5 в системе информационного обмена. Наконец, для нового способа можно назначать имена, указывающие на запомненные элементы данных, в каждом контексте отдельно.

Таким образом, обращение к запомненным элементам данных как части команды, принимаемой с одного из блоков обработки данных 4, 5, можно делать весьма эффективным: из-за очень небольшого количества элементов данных и небольшого количества отдельных операций, которые используются в современной практике интеллектуальных карточек, в каждом условии отдельно необходимо только несколько двоичных разрядов для кодирования имени и области команд. При аналогичных условиях доступа способы их верификации и криптографические операции, пригодные для этой цели в существующих интеллектуальных карточках, будут весьма ограничены по количеству, и их можно выразить очень эффективно в двухярусной иерархии описаний контекста взаимодействия 19(1)..., заключенных в описании прикладной задачи 18.

Формула изобретения:

1. Система информационного обмена, содержащая по меньшей мере один портативный блок обработки данных (5), включающий в себя средство передачи информации (14), выполненное с возможностью информационного обмена с другим портативным блоком обработки данных (4), средство обработки (15) и запоминающее средство (16), средство обработки соединено со средством передачи информации (14) и с запоминающим средством (16), причем запоминающее средство содержит первую область с управляющей программой (17), дополнительно содержит вне первой области вторую область, содержащую описания возможных режимов связи между блоками обработки в виде контекстов взаимодействия (19(1)... 19(m), причем упомянутая вторая область запоминающего средства конфигурирована в соответствии со следующей структурой данных:

a) набор основных примитивов связи (A(1)...), которые принимаются в качестве команд при их приеме первым блоком обработки данных (5), осуществляющим связь по меньшей мере с одним вторым блоком обработки данных (4), причем упомянутые примитивы по меньшей мере включают в себя примитив, используемый для избирательного ввода одного из упомянутых контекстов взаимодействия (19(1)...);

b) набор процедурных описаний (C(1)...), определяющих действия, подлежащие выполнению первым блоком обработки данных (5) в ответ на каждый из принимаемых примитивов связи (A(1)...), по меньшей мере включающий в себя первое процедурное описание, подлежащее выполнению при активировании контекста взаимодействия, и последнее процедурное описание, подлежащее выполнению непосредственно перед деактивированием контекста взаимодействия;

c) возможно незаполненный набор элементов данных (H(1)...), либо постоянно хранящихся в запоминающем средстве, либо вычисляемых, которые доступны для использования при выполнении процедур, определяемых в процедурных описаниях (C(1)...), при использовании и доступе к упомянутым элементам данных;

d) возможно незаполненный первый набор ссылок (r(1), r(2), r(3)) на элементах данных (H(1)...), причем эти ссылки первого набора (r(1), r(2), r(3)) связаны с процедурными описаниями(C(1)...), так что упомянутые элементы данных доступны для использования при выполнении процедур, определенных в процедурных описаниях (C(1)...);

e) возможно незаполненный второй набор ссылок (r(4), r(5), r(6)) на элементах данных (H(1). . .), причем ссылки второго набора (r(4), r(5), r(6)) связаны с процедурными описаниями (C(4)...) дополнительных контекстов взаимодействия, так что элементы данных доступны для использования, когда выполняются процедуры, определенные в процедурных описаниях (C(1)...) дополнительных контекстов взаимодействия;

f) возможно незаполненную первую таблицу данных (B(1)...), содержащую возможно упорядоченный третий набор ссылок (n(1)...) на упомянутые элементы данных (H(1)...), причем третий набор ссылок доступен в качестве указателей ссылок четвертого набора ссылок (W(1)...), причем ссылки четвертого набора ссылок (W(1)...) представляют собой часть упомянутых примитивов связи (A(1). . .), при этом элементы данных предназначены для использования процедурными описаниями

RU 2148856 C1

RU 2148856 C1

(C(1)...), связанными с примитивами связи (A(1)...);

g) первый набор условий доступа, связанных с элементами данных (H(1)...), на которые ссылаются в связи с первым набором (r(1), r(2), r(3)) и втором набором (r(4), r(5), r(6)) ссылок на элементы данных;

h) второй набор условий доступа, связанных с третьим набором ссылок (n(1)...) в первой таблице данных (B(1)...).

2. Система информационного обмена по п.1, отличающаяся тем, что запоминающее средство (16) дополнительно содержит по меньшей мере два контекста взаимодействия (19(1)...19(m), по меньшей мере одно описание прикладной задачи (18(1). . .) и элемент памяти (20) для запоминания ссылки на контекст взаимодействия, находящийся в данный момент в действии, причем каждое описание прикладной задачи содержит: a) таблицу данных, содержащую в себе ссылки (E(1). . .) на элементы данных, и эти ссылки могут быть доступными для двух или более контекстов взаимодействия (19)...) и могут расширяться с помощью дополнительных элементов данных; b) дополнительный набор условий доступа, связанных с упомянутыми ссылками (E(1)...) или с упомянутыми дополнительными элементами данных и определяющие ограничения использования.

3. Система информационного обмена по п.2, отличающаяся тем, что каждое описание прикладной задачи (18(1)...) содержит также библиотеку процедур, содержащую блоки исполняемых кодов (A(1)...), которые могут использоваться процедурными описаниями (C(1)...) каждого контекста взаимодействия, связанного с каждым из упомянутых описаний прикладных задач (18(1)...).

4. Система информационного обмена по п.2, отличающаяся тем, что запоминающее средство содержит по меньшей мере два описания прикладных задач (18(1). . .) и блоки исполняемых кодов (G(1), которые могут использоваться процедурными описаниями (C(1)...) каждого контекста взаимодействия (19(1)...) в каждом описании прикладной задачи (18(1)...) или каждым блоком исполняемых кодов (F(1)...) каждой библиотеки процедур в каждом описании прикладной задачи (18(1)...).

5. Система информационного обмена по п.3, отличающаяся тем, что блоки исполняемого кода в библиотеке процедур модифицированы путем включения спецификации использования их операционных параметров в классы, относящиеся к определяющим признакам, касающимся элементов данных, которые можно прогонять при вычислениях как действительные значения, причем вычисление производится только при согласовании признаков данных и классов параметров.

6. Система информационного обмена по п.2, отличающаяся тем, что управляющая программа (17) содержит ссылку на устанавливаемый по умолчанию контекст взаимодействия, который используется для инициализации элемента (20) запоминающего устройства, хранящего ссылку на контекст взаимодействия, находящийся в настоящий момент в действии, для выполнения конечного действия после выявления внутренней несовместимости при возврате в нормальное состояние операции или всякий раз, когда управляющая программа (17) активирована и не определен в явном виде контекст взаимодействия примитивом связи, принимаемым от второго блока обработки данных (4).

7. Система информационного обмена по п.1, отличающаяся тем, что запоминающее средство (16) дополнительно содержит контекст взаимодействия, предназначенный для включения в себя персональных идентификационных номеров, а управляющая программа (17) обеспечивает проверку персональных идентификационных номеров, обеспечиваемых пользователем системы информационного обмена.

8. Система информационного обмена по п.2, отличающаяся тем, что каждое описание прикладной задачи (18(1)...) содержит таблицу числовых значений, которая составлена для обеспечения идентификаторов всех контекстов взаимодействия (19(1)...) и содержит по меньшей мере первое числовое значение, указывающее тип прикладной задачи, второе числовое значение, указывающее однозначную идентификацию поставщика прикладной задачи, третье числовое значение, указывающее характер описания прикладной задачи (18(1)...), и дополнительные числа, каждое из которых однозначно указывает на один контекст взаимодействия (19(1)...), связанный с описанием прикладной задачи.

9. Система информационного обмена по п.1, отличающаяся тем, что средство передачи данных (14) обеспечивает структурирование обмена данными в виде блоков данных, содержащих по меньшей мере две части, причем первая часть представляет собой данные, определенные в качестве операционных в том смысле, что они используются в качестве влияющих на характер операций, осуществляемых по команде, как указано примитивом связи, или данные, получающиеся в результате выполненных операций, вторая часть данных определена в качестве защиты в том смысле, что она используется для выявления соответствия выполнения операции или приемлемости данных в операционной части, подлежащих использованию в операции, или для подтверждения завершения операции или правильности полученных в результате данных.

10. Система информационного обмена по п.9, отличающаяся тем, что управляющая программа (17) приспособлена для выполнения при принятии примитива связи операций, определяемых в текущем контексте взаимодействия (19(1)...) каждой операции как части заранее установленной и фиксированной последовательности действий, каждое из которых определяется отдельно в виде части процедурного описания, связанного с принимаемым примитивом связи, и эти действия включают в себя по меньшей мере следующие действия:

a) разрешение использования примитива связи;

b) декодирование операционных данных или какой-либо их части;

c) выполнение команды с любыми входными данными;

RU 2148856 C1

RU 2148856 C1

d) кодирование любых операционных данных, полученных в результате выполнения какой-либо операции;

e) вычисление подтверждения завершения любого выполняемого действия или правильности полученных в результате данных, подлежащих использованию при вычислениях надежности.

11. Система информационного обмена по п.1, отличающаяся тем, что блок обработки данных (5) формирует случайный номер транзакции при инициализировании передачи данных, которые служат в качестве основы для криптографических вычислений.

12. Система информационного обмена по п.1, отличающаяся тем, что одному примитиву связи назначается специальное значение, которое всегда будет интерпретироваться как запрос на ввод нового контекста взаимодействия (19(1)...).

13. Система информационного обмена по п.1, отличающаяся тем, что она содержит дополнительный блок обработки данных (4), содержащий в себе те же самые элементы, что и блок обработки данных (5), который может дополнительно содержать в своем запоминающем устройстве программный интерфейс (10) прикладных задач, который состоит из кода программы, предназначенного для обеспечения возможности выполнения дополнительных компьютерных программ, для управления пользователем последовательностью обмениваемых примитивов связи или для воздействия на передаваемые в них данные или для обучения или дополнительной обработки данных, принимаемых при обмене.

14. Система информационного обмена по п.13, отличающаяся тем, что примитив, используемый для ввода определенного контекста взаимодействия (19(1)...), содержит числовые значения, подлежащие использованию при вычислениях надежности в последующих информационных обменах, первое произвольное значение, генерируемое одним из блоков обработки, и второе значение, служащее для идентификации одного блока обработки.

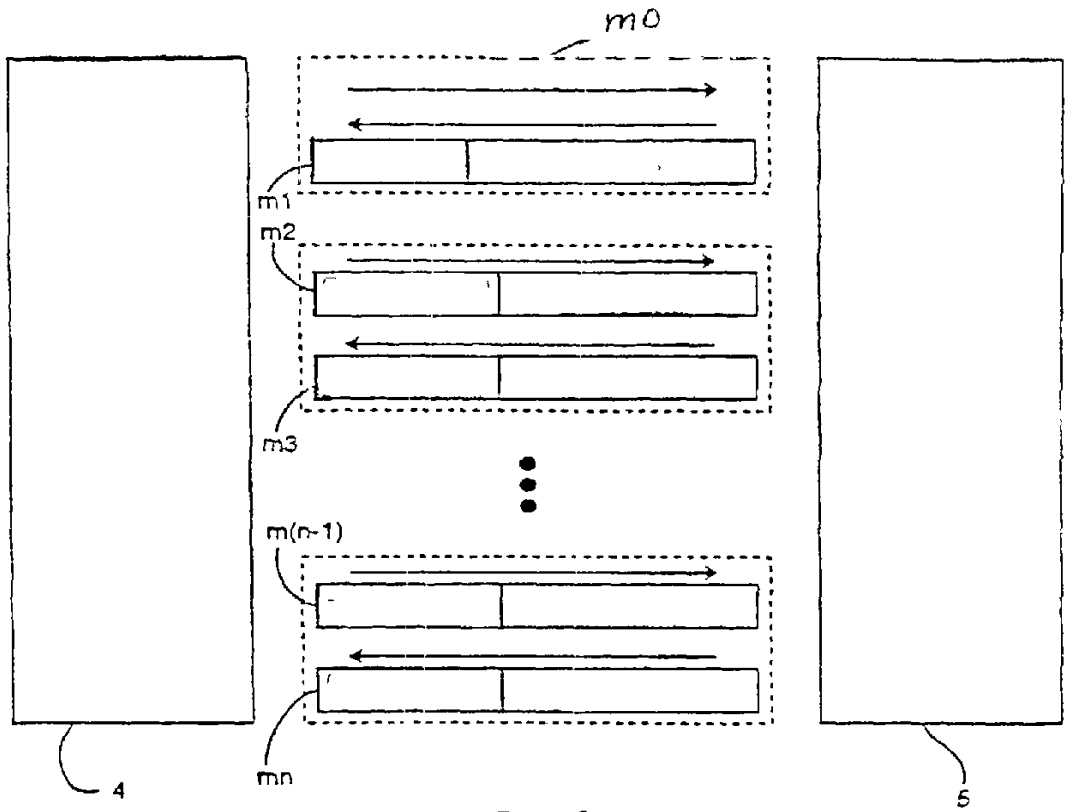
15. Система информационного обмена по п.13, отличающаяся тем, что каждый примитив связи состоит из двух или более числовых значений, причем первое значение используется для указания на процедурное описание действия, связанного с примитивом связи, второе значение составлено из постоянного числа двоичных значений, каждое из которых интерпретируется управляющей программой (12, 17) как ссылка на единственный элемент данных.

16. Система информационного обмена по п.13, отличающаяся тем, что каждый примитив связи состоит из двух или более числовых значений, причем первое значение используется для указания на процедурное описание действия, связанного с примитивом связи, второе значение используется для определения, который из элементов данных, пригодный для внешней ссылки в активированном контексте взаимодействия (19(1). ..), должен использоваться при выполнении ответственных действий таким образом, чтобы выбирался конкретный элемент данных, если он содержит значение, которое согласуется с упомянутым вторым значением.

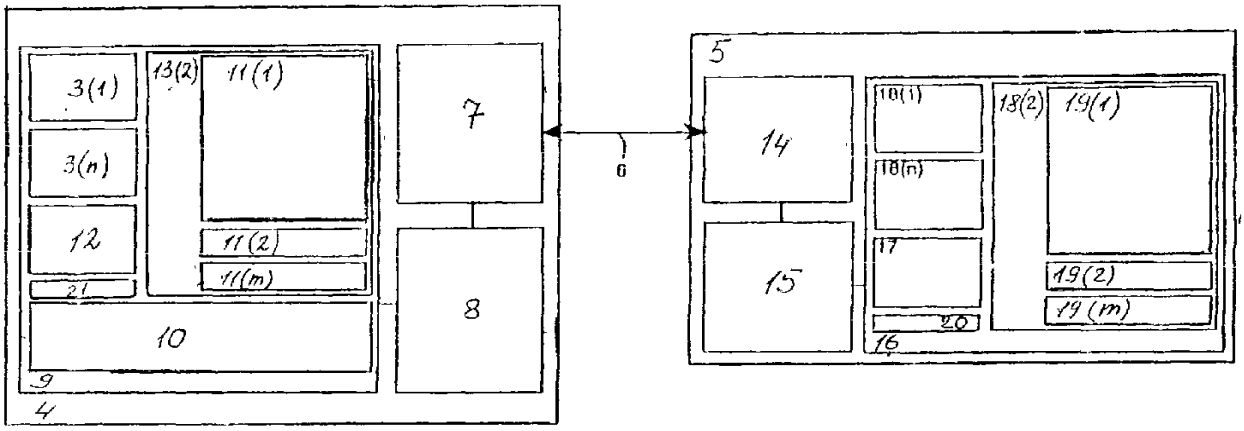
17. Система информационного обмена по п.13, отличающаяся тем, что каждый примитив связи состоит из двух или более числовых значений, причем первое значение используется для указания на процедурное описание действия, связанного с примитивом связи, второе значение состоит из ряда двоичных значений, которым с помощью управляющей программы (12, 17) назначены конкретные значения, подлежащие использованию при интерпретировании форматов данных в примитиве связи и при выполнении ответных действий.

RU 2148856 C1

RU 2148856 C1



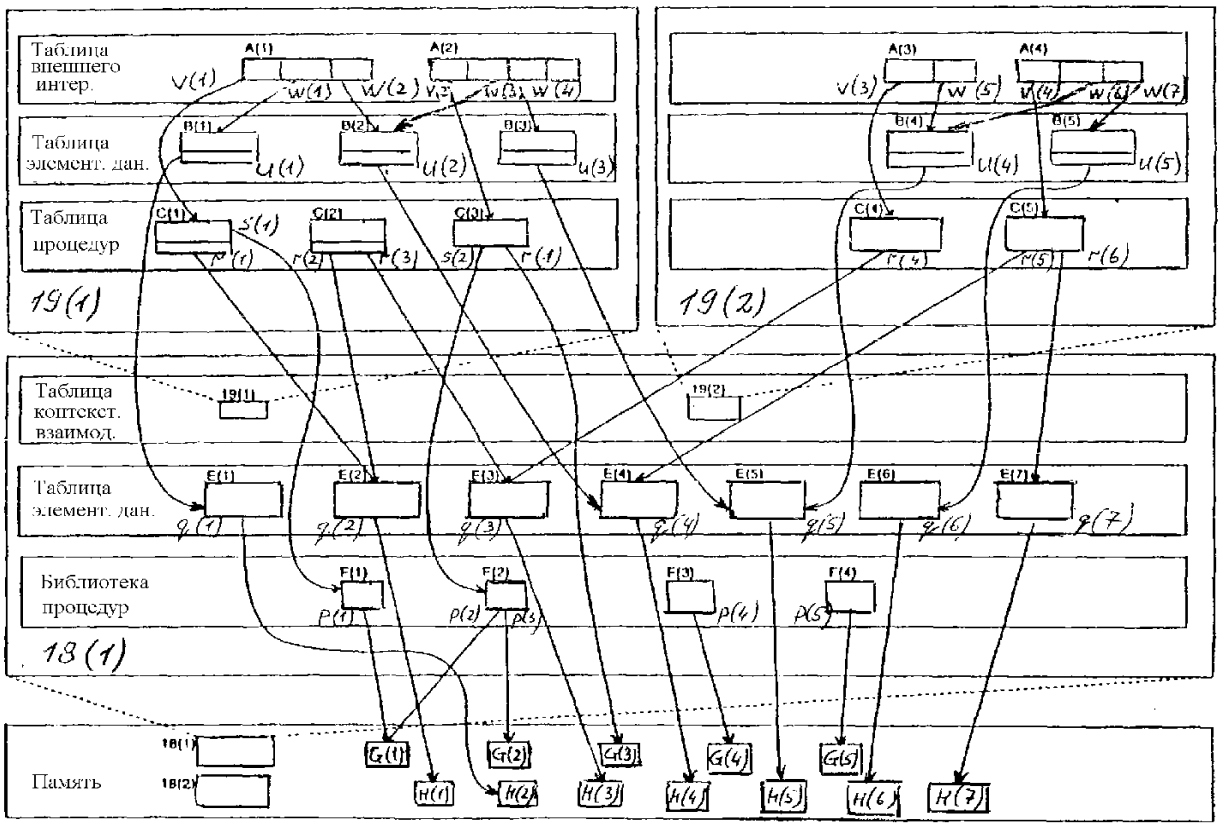
Фиг.2



Фиг.3

RU 2148856 C1

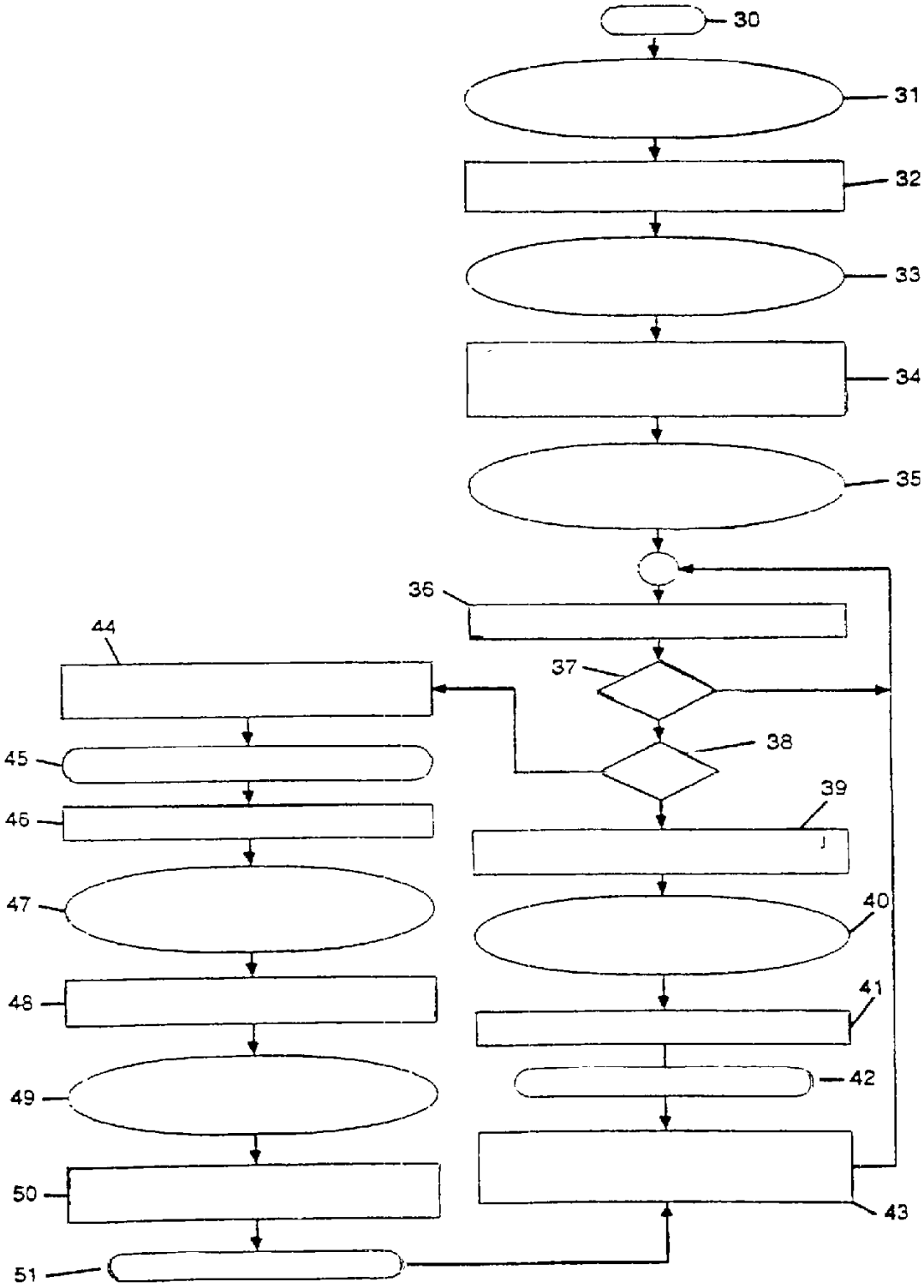
RU 2148856 C1



Фиг.4

RU 2148856 C1

RU 2148856 C1



Фиг.5

RU 2148856 C1